Linnéuniversitetet
Kalmar Växjö

Report

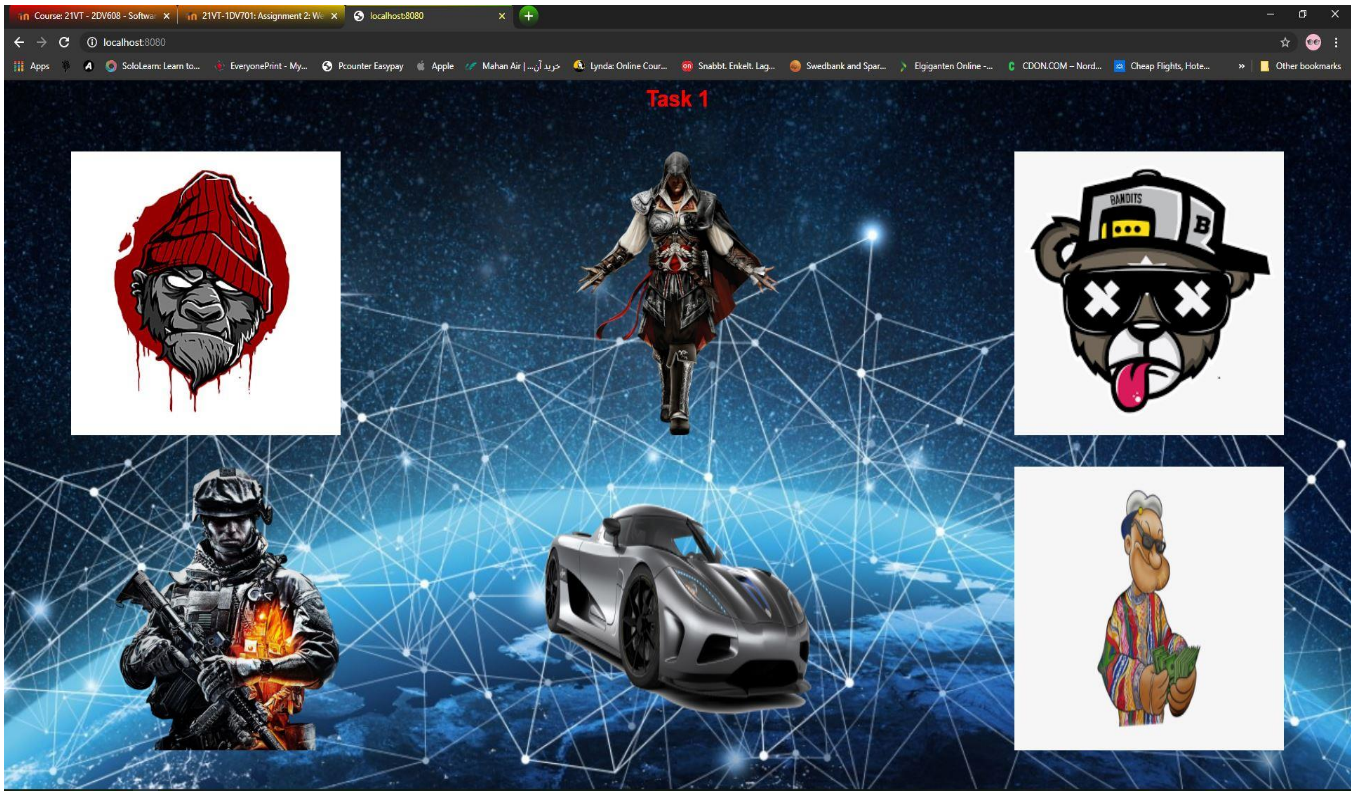
Assignment 2

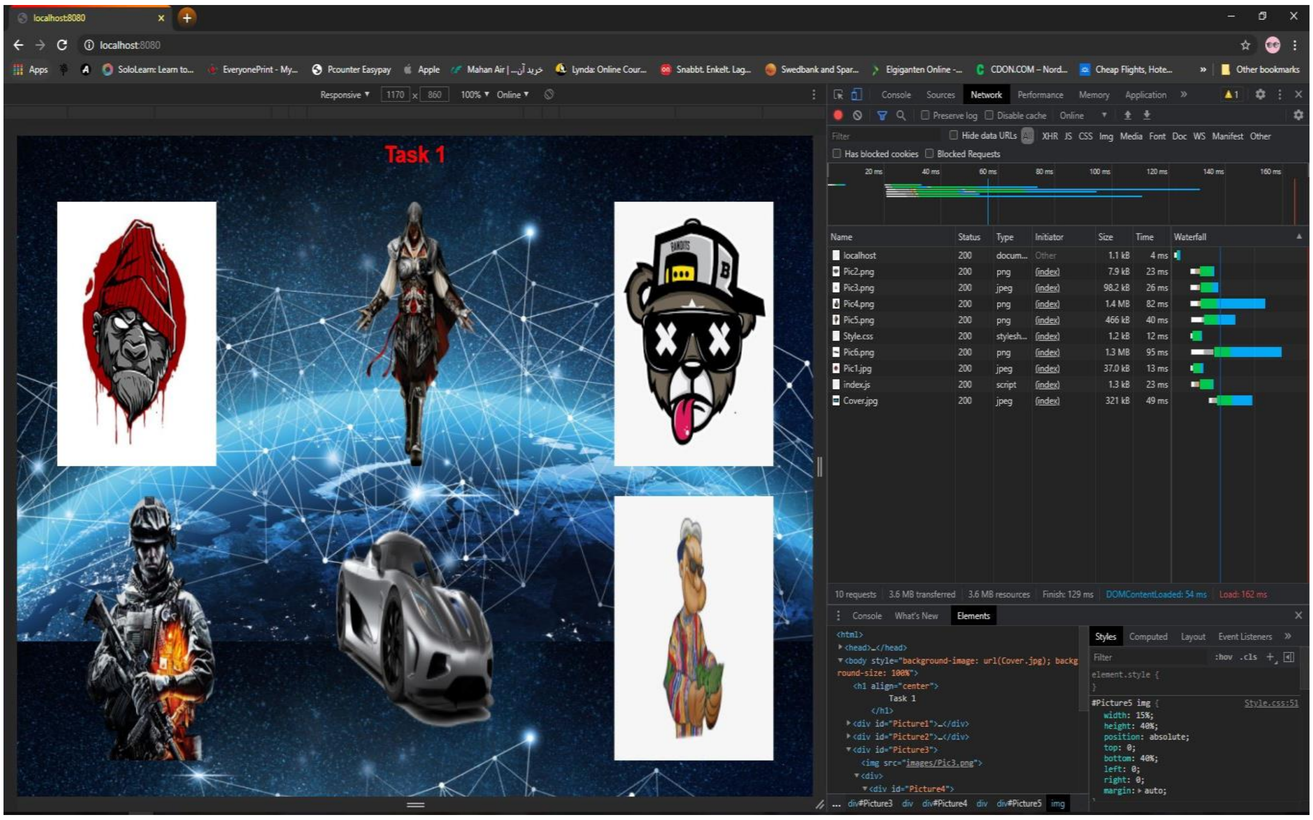
1DV701

Author: Mohammadali Rashidfarokhi & Nima Safavi
Semester: Spring 2020
Email: *mr223jp@student.lnu.se*
ns222tv@student.lnu.se

Contents

1 Problem 1




localhost:8080
Task 1
Responsive 1170 x 860 100% Online
Console Sources Network Performance Memory Application
Preserve log Disable cache Online
Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other
Has blocked cookies Blocked Requests

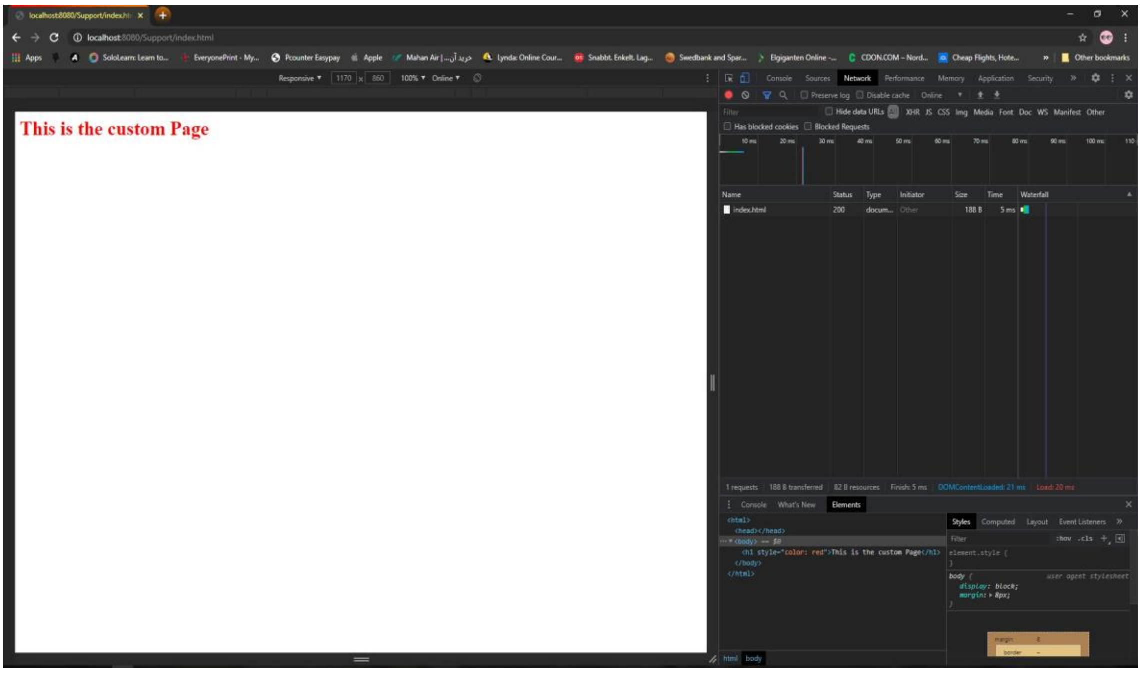
Name	Status	Type	Initiator	Size	Time
localhost	200	docum...	Other	1.1 kB	4 ms
Pic2.png	200	png	(index)	7.9 kB	23 ms
Pic3.png	200	jpeg	(index)	98.2 kB	26 ms
Pic4.png	200	png	(index)	1.4 MB	82 ms
Pic5.png	200	png	(index)	466 kB	40 ms
Style.css	200	stylesh...	(index)	1.2 kB	12 ms
Pic6.png	200	png	(index)	1.3 MB	95 ms
Pic1.jpg	200	jpeg	(index)	37.0 kB	13 ms
index.js	200	script	(index)	1.3 kB	23 ms
Cover.jpg	200	jpeg	(index)	321 kB	49 ms

10 requests | 3.6 MB transferred | 3.6 MB resources | Finish: 129 ms | DOMContentLoaded: 54 ms | Load: 162 ms

```
<html>
 ▸<head>…</head>
 ▾<body style="background-image: url(Cover.jpg); background-size: 100%">
    <h1 align="center">
          Task 1
      </h1>
   ▸<div id="Picture1">…</div>
   ▸<div id="Picture2">…</div>
   ▾<div id="Picture3">
      <img src="images/Pic3.png">
     ▾<div>
      ▾<div id="Picture4">
```

```
element.style {
}
#Picture5 img {                                   Style.css:51
   width: 15%;
   height: 40%;
   position: absolute;
   top: 0;
   bottom: 40%;
   left: 0;
   right: 0;
   margin: ▸ auto;
```

... div#Picture3 div div#Picture4 div div#Picture5 img


localhost:8080/Support/index.html

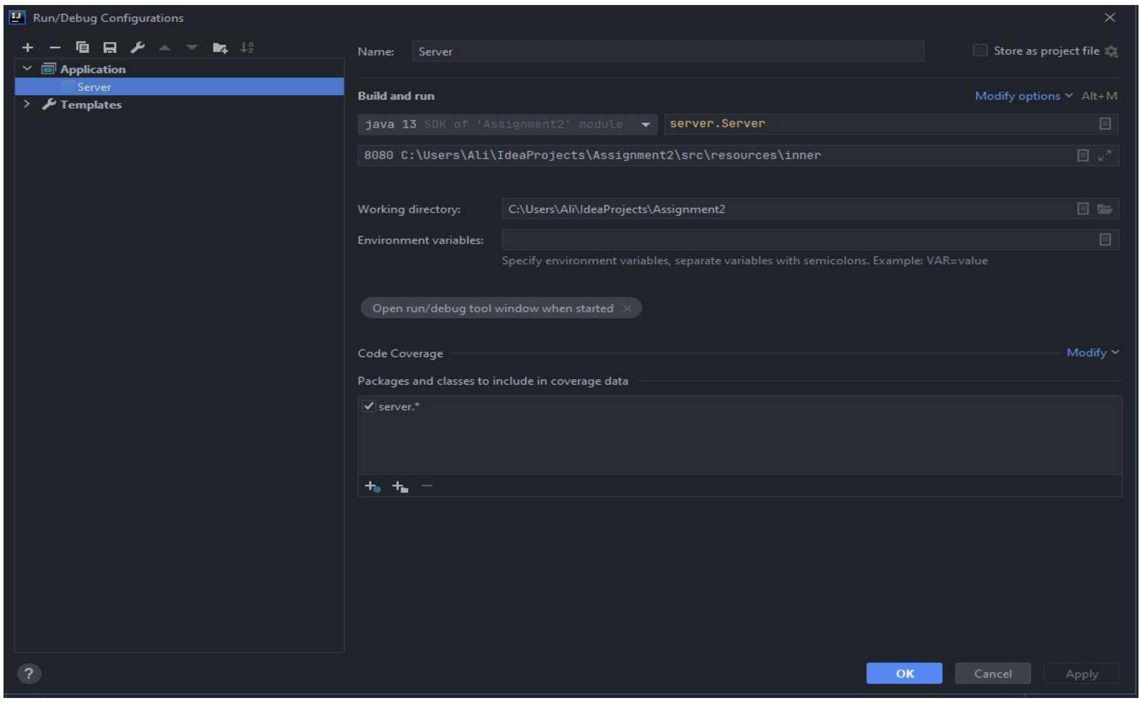
This is the custom Page

Name	Status	Type	Initiator	Size	Time
index.html	200	docum...	Other	188 B	5 ms

1 requests | 188 B transferred | 82 B resources | Finish: 5 ms | DOMContentLoaded: 21 ms | Load: 20 ms

```
<html>
  <head></head>
  <body> == $0
    <h1 style="color: red">This is the custom Page</h1>
  </body>
</html>
```

```
element.style {
}
body {                      user agent stylesheet
  display: block;
  margin: ▸ 8px;
}
```


Run/Debug Configurations

Name: Server

Build and run

java 13 SDK of 'Assignment2' module | server.Server

8080 C:\Users\Ali\IdeaProjects\Assignment2\src\resources\inner

Working directory: C:\Users\Ali\IdeaProjects\Assignment2

Environment variables:

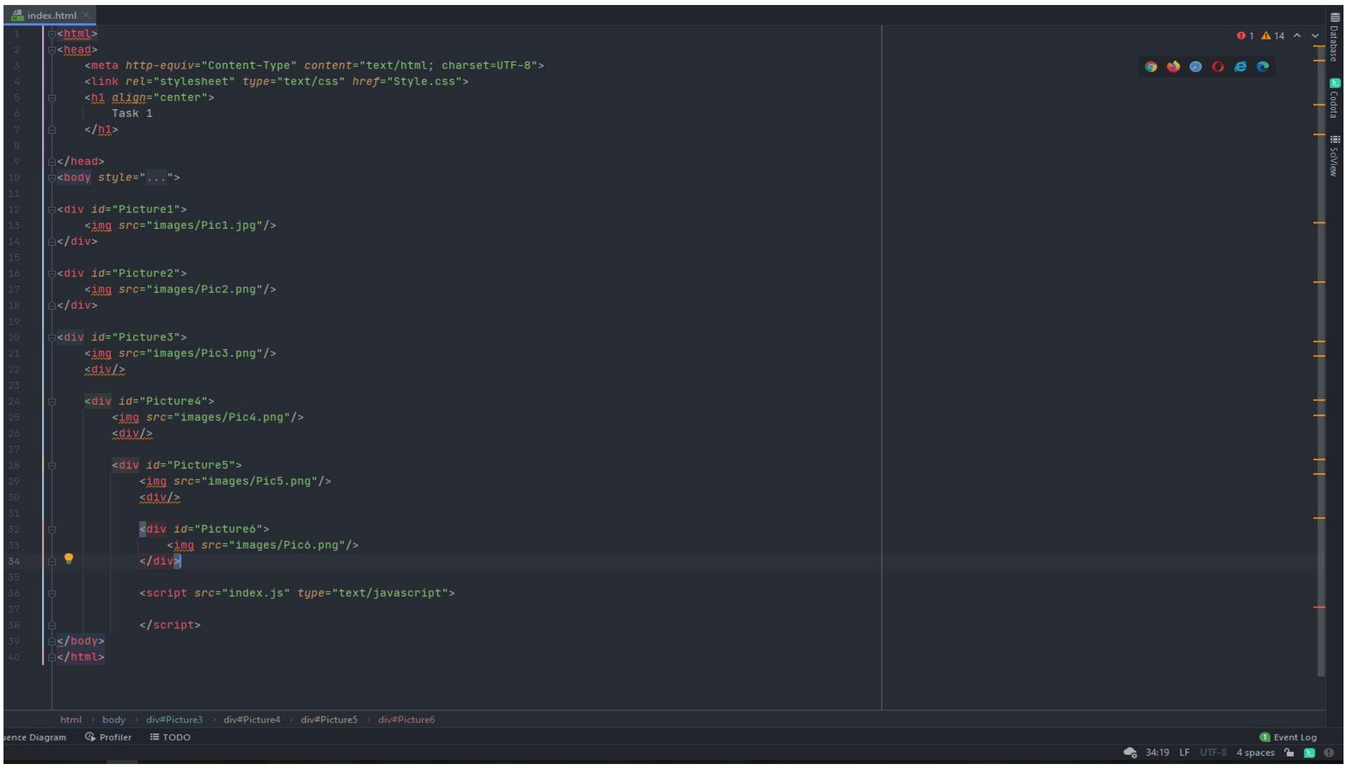
Specify environment variables, separate variables with semicolons. Example: VAR=value

Open run/debug tool window when started

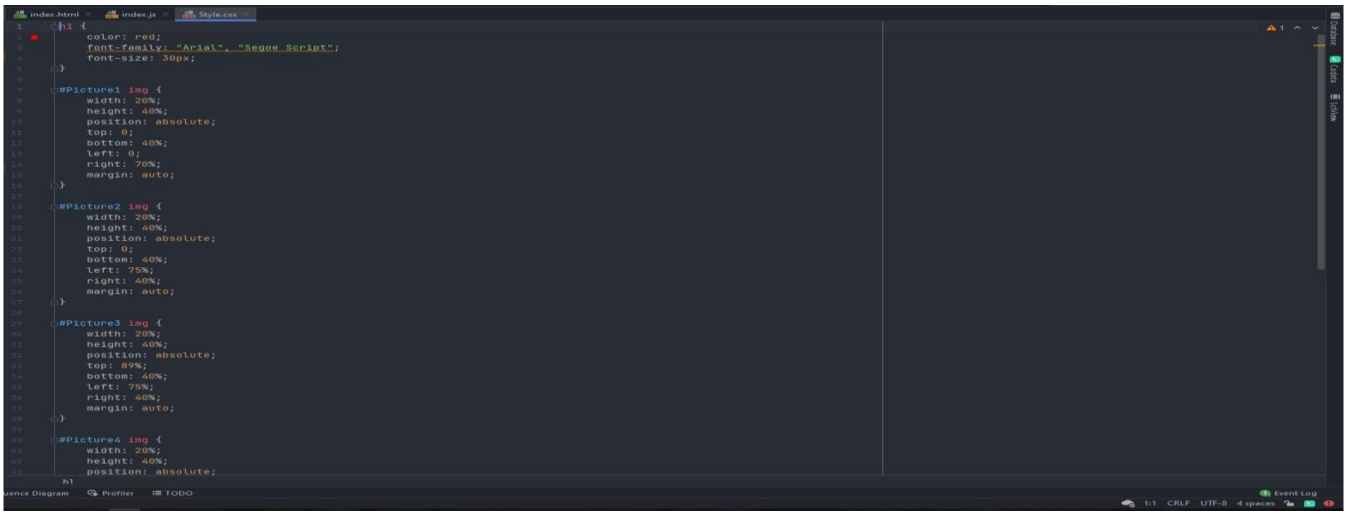
Code Coverage

Packages and classes to include in coverage data

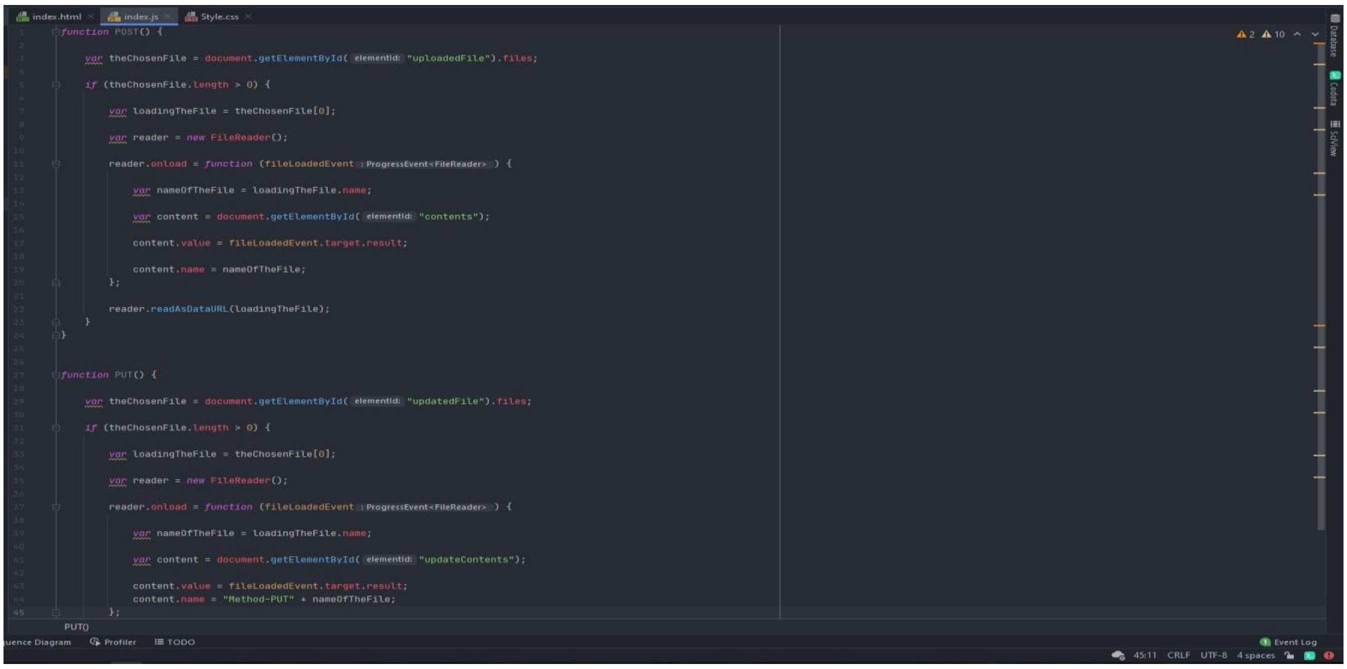
✓ server.*



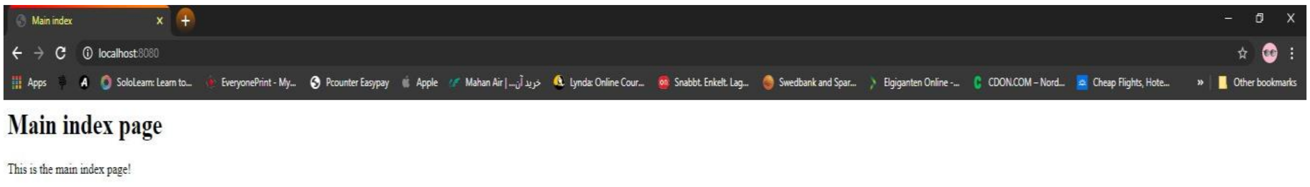
```html
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" type="text/css" href="Style.css">
    <h1 align="center">
        Task 1
    </h1>

</head>
<body style="...">

<div id="Picture1">
    <img src="images/Pic1.jpg"/>
</div>

<div id="Picture2">
    <img src="images/Pic2.png"/>
</div>

<div id="Picture3">
    <img src="images/Pic3.png"/>
    <div/>

    <div id="Picture4">
        <img src="images/Pic4.png"/>
        <div/>

        <div id="Picture5">
            <img src="images/Pic5.png"/>
            <div/>

            <div id="Picture6">
                <img src="images/Pic6.png"/>
            </div>

            <script src="index.js" type="text/javascript">

            </script>
</body>
</html>
```



```css
h1 {
    color: red;
    font-family: "Arial", "Segoe Script";
    font-size: 30px;
}

#Picture1 img {
    width: 20%;
    height: 40%;
    position: absolute;
    top: 0;
    bottom: 40%;
    left: 0;
    right: 70%;
    margin: auto;
}

#Picture2 img {
    width: 20%;
    height: 40%;
    position: absolute;
    top: 0;
    bottom: 40%;
    left: 75%;
    right: 40%;
    margin: auto;
}

#Picture3 img {
    width: 20%;
    height: 40%;
    position: absolute;
    top: 89%;
    bottom: 40%;
    left: 75%;
    right: 40%;
    margin: auto;
}

#Picture4 img {
    width: 20%;
    height: 40%;
    position: absolute;
```

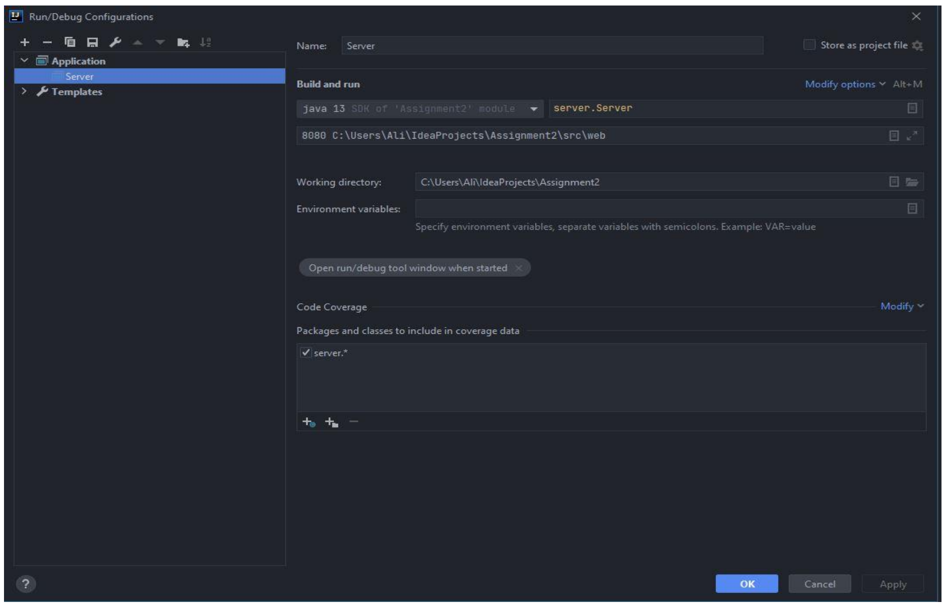


```javascript
function POST() {

    var theChosenFile = document.getElementById("uploadedFile").files;

    if (theChosenFile.length > 0) {

        var loadingTheFile = theChosenFile[0];

        var reader = new FileReader();

        reader.onload = function (fileLoadedEvent) {

            var nameOfTheFile = loadingTheFile.name;

            var content = document.getElementById("contents");

            content.value = fileLoadedEvent.target.result;

            content.name = nameOfTheFile;
        };

        reader.readAsDataURL(loadingTheFile);
    }
}


function PUT() {

    var theChosenFile = document.getElementById("updatedFile").files;

    if (theChosenFile.length > 0) {

        var loadingTheFile = theChosenFile[0];

        var reader = new FileReader();

        reader.onload = function (fileLoadedEvent) {

            var nameOfTheFile = loadingTheFile.name;

            var content = document.getElementById("updateContents");

            content.value = fileLoadedEvent.target.result;
            content.name = "Method-PUT" + nameOfTheFile;
        };
```



Main index page

This is the main index page!



Name: Server

Store as project file

Build and run

```
java 13 SDK of 'Assignment2' module    server.Server
8080 C:\Users\Ali\IdeaProjects\Assignment2\src\web
```

Working directory: C:\Users\Ali\IdeaProjects\Assignment2

Environment variables:

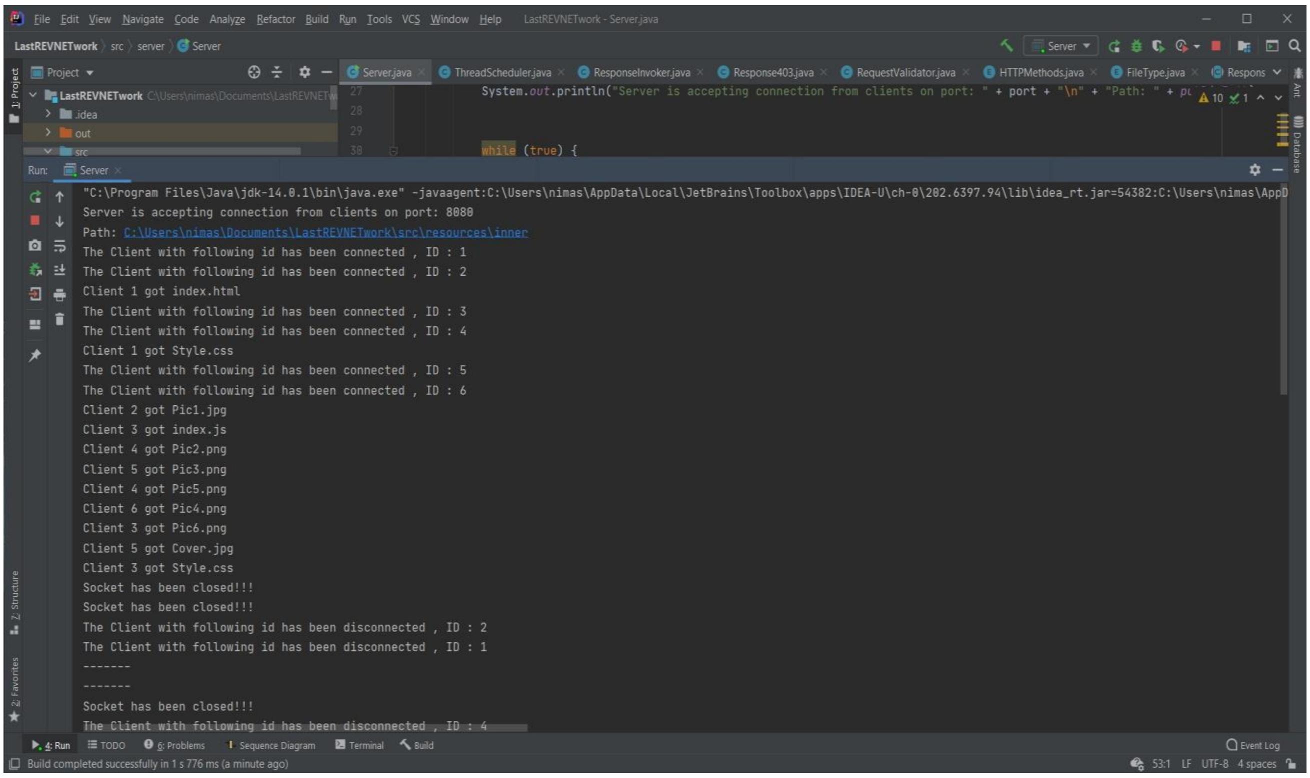
Specify environment variables, separate variables with semicolons. Example: VAR=value

Open run/debug tool window when started

Code Coverage

Packages and classes to include in coverage data

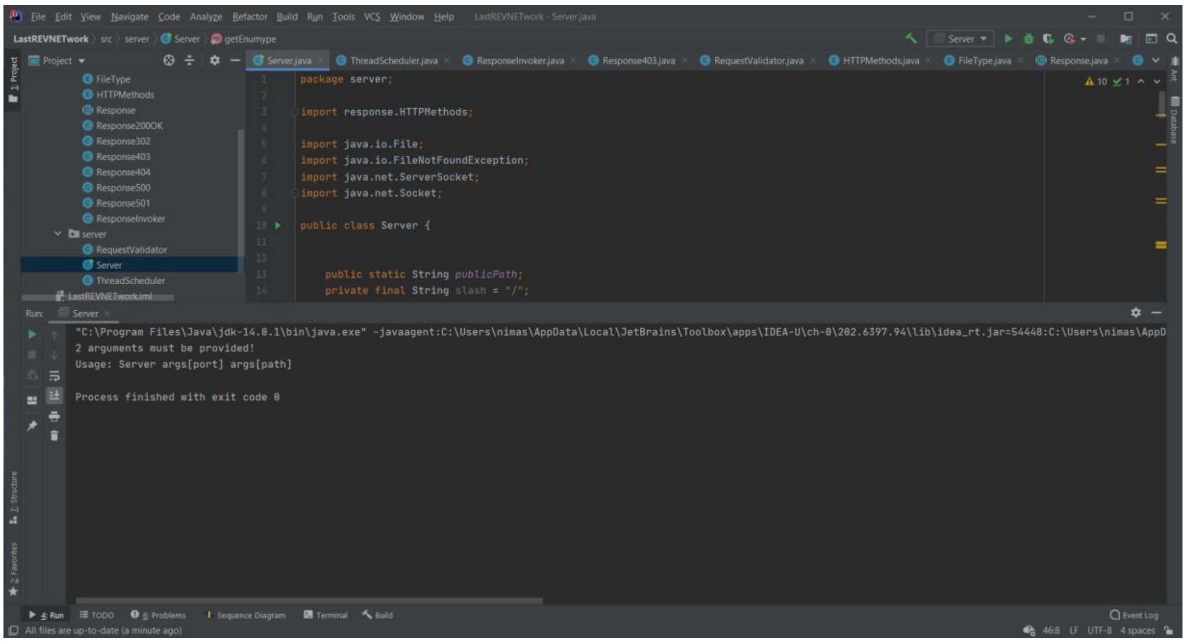
✓ server.*



```
System.out.println("Server is accepting connection from clients on port: " + port + "\n" + "Path: " + pc

while (true) {
```

```
"C:\Program Files\Java\jdk-14.0.1\bin\java.exe" -javaagent:C:\Users\nimas\AppData\Local\JetBrains\Toolbox\apps\IDEA-U\ch-0\202.6397.94\lib\idea_rt.jar=54382:C:\Users\nimas\AppD
Server is accepting connection from clients on port: 8080
Path: C:\Users\nimas\Documents\LastREVNETwork\src\resources\inner
The Client with following id has been connected , ID : 1
The Client with following id has been connected , ID : 2
Client 1 got index.html
The Client with following id has been connected , ID : 3
The Client with following id has been connected , ID : 4
Client 1 got Style.css
The Client with following id has been connected , ID : 5
The Client with following id has been connected , ID : 6
Client 2 got Pic1.jpg
Client 3 got index.js
Client 4 got Pic2.png
Client 5 got Pic3.png
Client 4 got Pic5.png
Client 6 got Pic4.png
Client 3 got Pic6.png
Client 5 got Cover.jpg
Client 3 got Style.css
Socket has been closed!!!
Socket has been closed!!!
The Client with following id has been disconnected , ID : 2
The Client with following id has been disconnected , ID : 1
-------
-------
Socket has been closed!!!
The Client with following id has been disconnected , ID : 4
```

Build completed successfully in 1 s 776 ms (a minute ago)



```java
package server;

import response.HTTPMethods;

import java.io.File;
import java.io.FileNotFoundException;
import java.net.ServerSocket;
import java.net.Socket;

public class Server {


    public static String publicPath;
    private final String slash = "/";
```

```
"C:\Program Files\Java\jdk-14.0.1\bin\java.exe" -javaagent:C:\Users\nimas\AppData\Local\JetBrains\Toolbox\apps\IDEA-U\ch-0\202.6397.94\lib\idea_rt.jar=54448:C:\Users\nimas\AppD
2 arguments must be provided!
Usage: Server args[port] args[path]

Process finished with exit code 0
```

1.1 Discussion

The above pictures will illustrate the connection between the server and client using port 8080. Based on the provided screenshot, it is visible that not only the server has the capability of serving HTML and PNG files but JPEG files as well.

Based on the provided screenshots, the program arguments will require two arguments which are port and path. The default port which was used in the above screenshots is 8080. Regarding the path argument, the provided path is pointing to the resources and inner. For example, the following path is the path from the host machine of the other group member.

“C:\Users\nimas\Documents\LastREVNETwork\src\resources\inner”

Apart from the main index that is visible in the first screenshot, we have used a custom index file that has been reached by using the following URL:

http://localhost:8080/Support/index.html

The implementation is supporting both CSS and javascript files. Consequntleley, their usages can be seen in the inspect / network section of the browser. Also, the design of the main index has been done by using the CSS file called *Style.css*.

Moving on, the output from the IntelliJ IDEA is stating that multiple clients have been connected and disconnected to the server. As a result, they are capable of receiving the files in different formats.

We were provided with a package called a web for testing the code. Therefore, as it has been shown above, we have changed one of our arguments (path) that is pointing to the web right now and it is working.

The implementation required us to handle some exceptions along the way. The following is a list of the exceptions that we have used in our implementation.

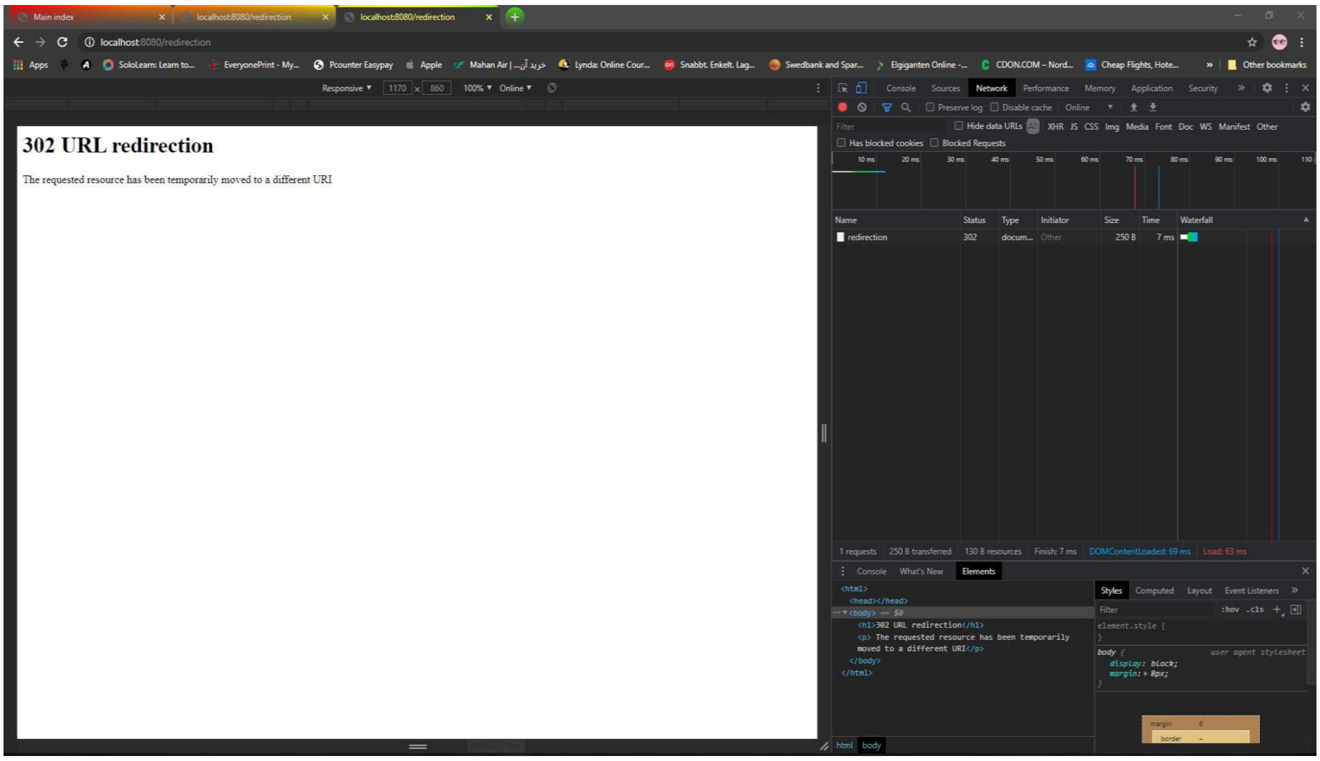
NoSuchFieldException	**RuntimeException**	**IoException**	**NullPointerException**
ArrayIndexOutOfBoundsException	**SecurityException**	**FileNotFoundException**	**NumberFormatException**
Exception			

The above exceptions have been used in our implementation. However, apart from their usage in general, mostly their usage can be seen in the second implementation.

Some of the basic error handlings also have been handled by using if statements and try-catch blocks. For example, the number of arguments is required to be 2. Otherwise, an error message will be shown indicating that the 2 arguments must be provided. This action is visible in the above screenshots.

All in all, the web server that we have designed is capable of supporting HTTP response known as 200 Ok.

2 Problem 2



localhost:8080/redirection

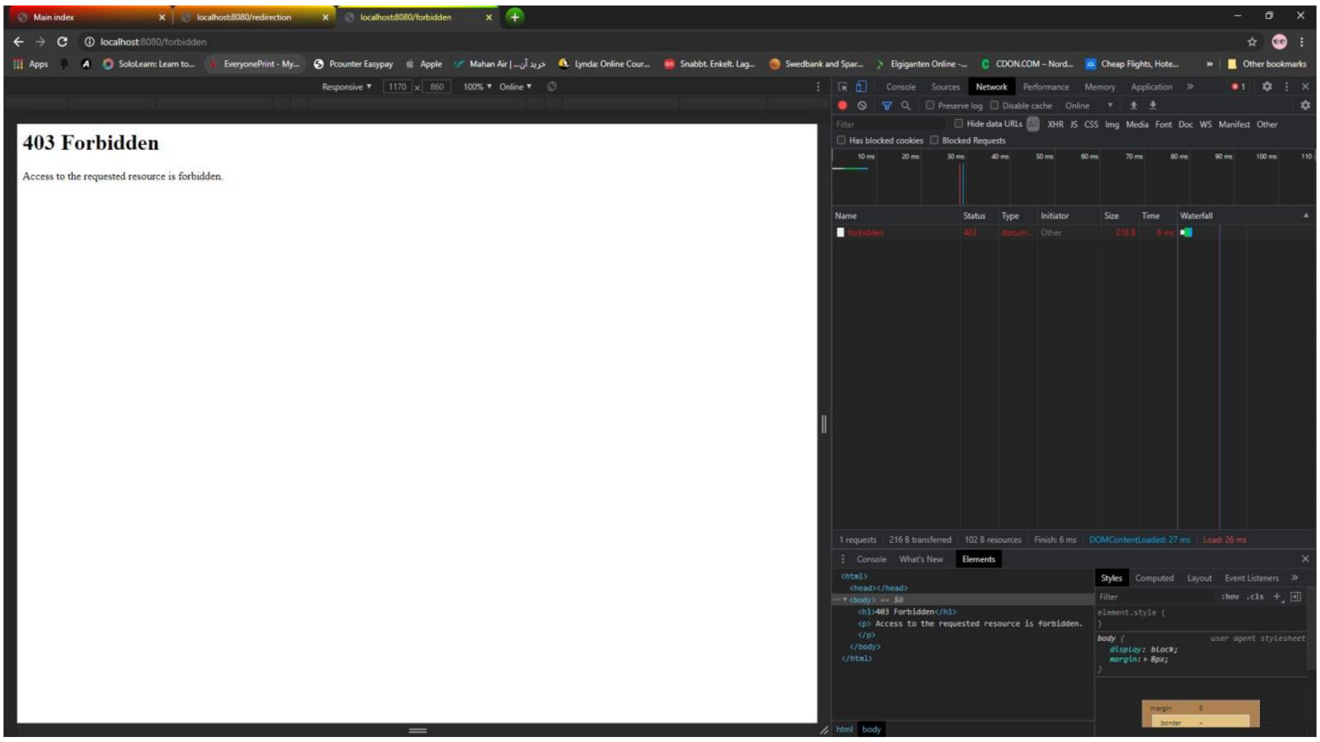
302 URL redirection

The requested resource has been temporarily moved to a different URI

```
Name          Status  Type     Initiator  Size   Time
redirection   302     docum…   Other      250 B  7 ms

1 requests | 250 B transferred | 130 B resources | Finish: 7 ms | DOMContentLoaded: 69 ms | Load: 63 ms
```

```
<html>
  <head></head>
  <body> == $0
    <h1>302 URL redirection</h1>
    <p> The requested resource has been temporarily
    moved to a different URI</p>
  </body>
</html>
```

```
body {                      user agent stylesheet
  display: block;
  margin: ▸ 8px;
}
```



localhost:8080/forbidden

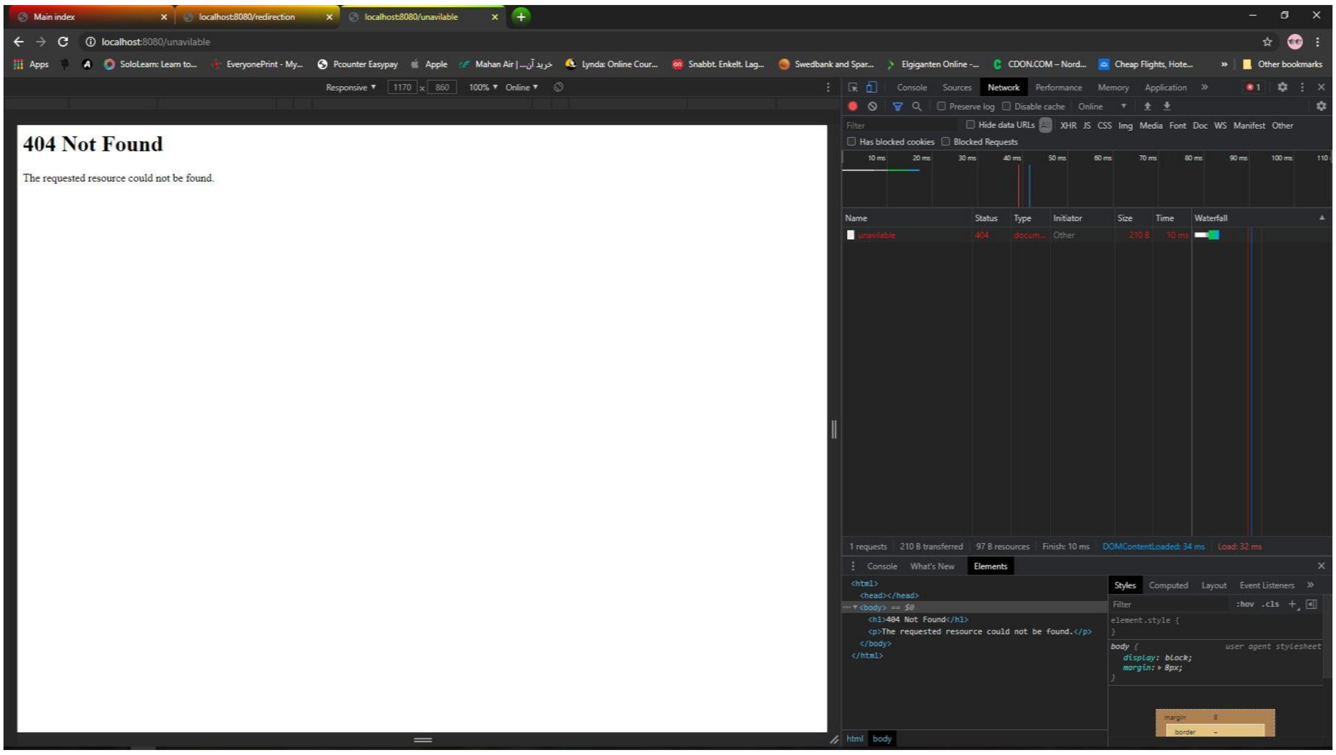
403 Forbidden

Access to the requested resource is forbidden.

```
Name        Status  Type     Initiator  Size   Time
forbidden   403     docum…   Other      216 B  6 ms

1 requests | 216 B transferred | 102 B resources | Finish: 6 ms | DOMContentLoaded: 27 ms | Load: 26 ms
```

```
<html>
  <head></head>
  <body> == $0
    <h1>403 Forbidden</h1>
    <p> Access to the requested resource is forbidden.
    </p>
  </body>
</html>
```

```
body {                      user agent stylesheet
  display: block;
  margin: ▸ 8px;
}
```


localhost:8080/unavilable

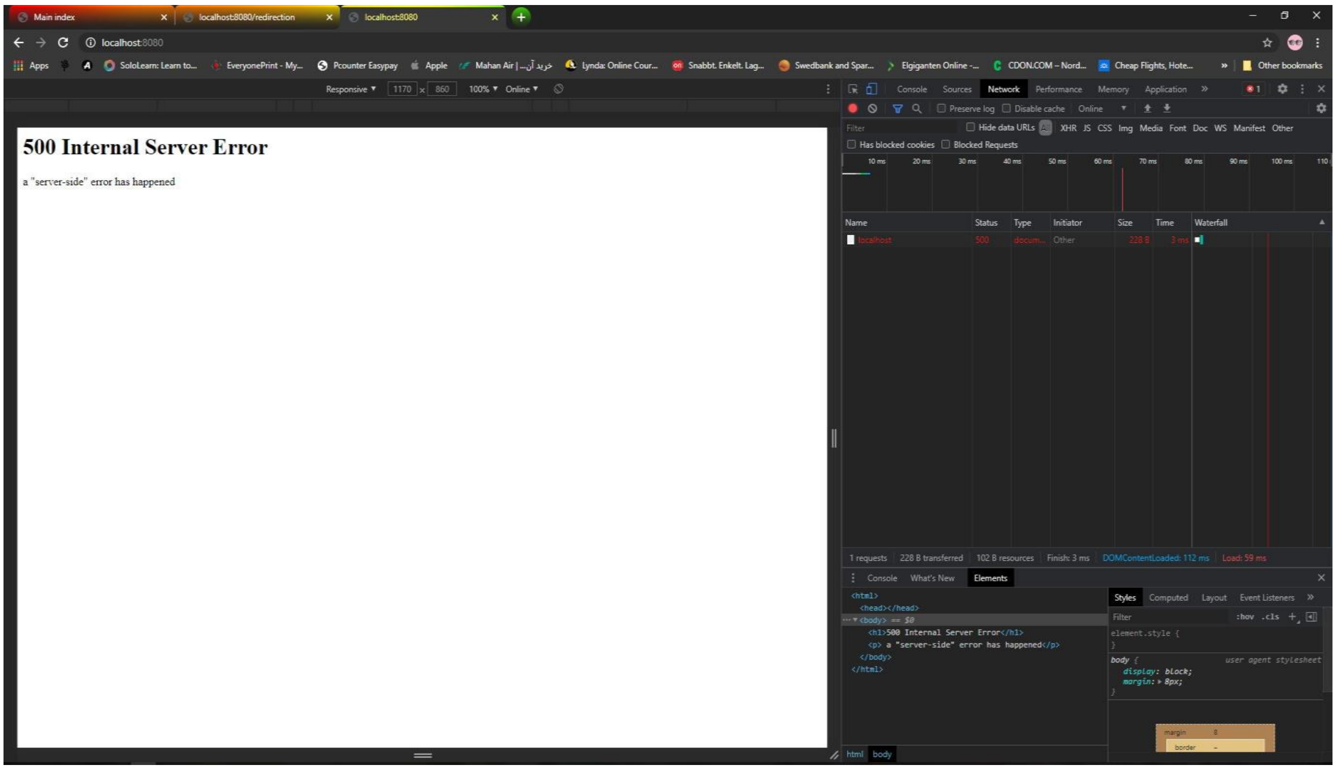
404 Not Found

The requested resource could not be found.

```
Name        Status  Type     Initiator  Size   Time   
unavilable  404     docum... Other      210 B  10 ms

1 requests | 210 B transferred | 97 B resources | Finish: 10 ms | DOMContentLoaded: 34 ms | Load: 32 ms
```

```
<html>
  <head></head>
··· ▼ <body> == $0
    <h1>404 Not Found</h1>
    <p>The requested resource could not be found.</p>
  </body>
</html>
```

```
element.style {
}
body {                      user agent stylesheet
  display: block;
  margin: ▸ 8px;
}
```


localhost:8080

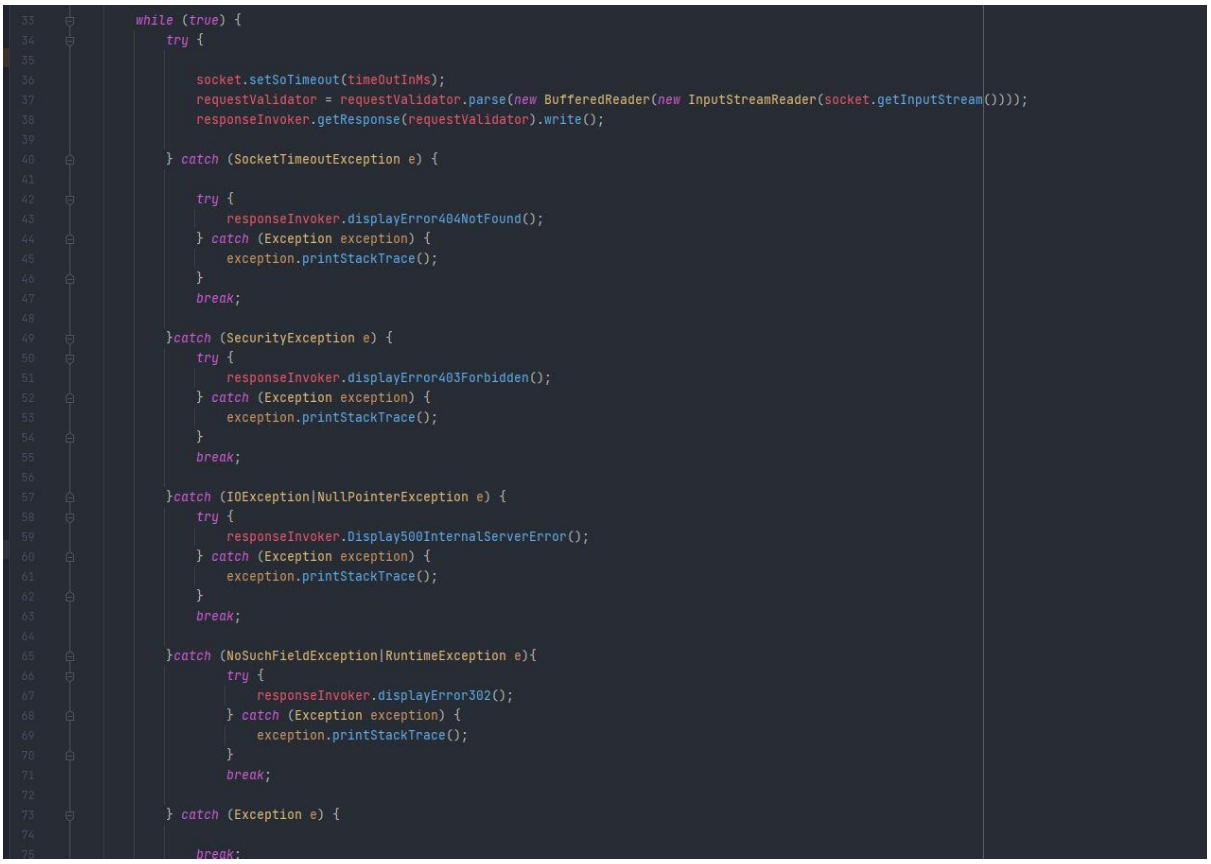
500 Internal Server Error

a "server-side" error has happened

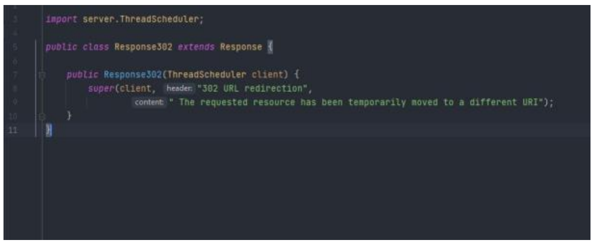
```
Name        Status  Type     Initiator  Size   Time   
localhost   500     docum... Other      228 B  3 ms

1 requests | 228 B transferred | 102 B resources | Finish: 3 ms | DOMContentLoaded: 112 ms | Load: 59 ms
```

```
<html>
  <head></head>
··· ▼ <body> == $0
    <h1>500 Internal Server Error</h1>
    <p> a "server-side" error has happened</p>
  </body>
</html>
```

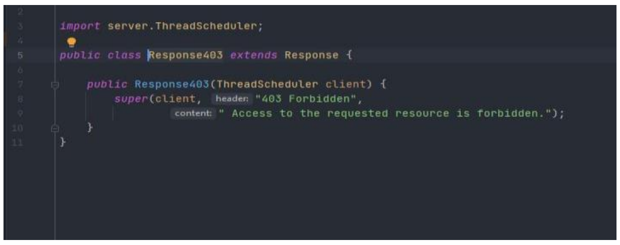
```
element.style {
}
body {                      user agent stylesheet
  display: block;
  margin: ▸ 8px;
}
```



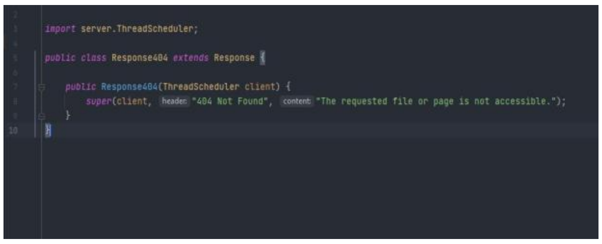
```
        while (true) {
            try {

                socket.setSoTimeout(timeOutInMs);
                requestValidator = requestValidator.parse(new BufferedReader(new InputStreamReader(socket.getInputStream())));
                responseInvoker.getResponse(requestValidator).write();

            } catch (SocketTimeoutException e) {

                try {
                    responseInvoker.displayError404NotFound();
                } catch (Exception exception) {
                    exception.printStackTrace();
                }
                break;

            }catch (SecurityException e) {
                try {
                    responseInvoker.displayError403Forbidden();
                } catch (Exception exception) {
                    exception.printStackTrace();
                }
                break;

            }catch (IOException|NullPointerException e) {
                try {
                    responseInvoker.Display500InternalServerError();
                } catch (Exception exception) {
                    exception.printStackTrace();
                }
                break;

            }catch (NoSuchFieldException|RuntimeException e){
                    try {
                        responseInvoker.displayError302();
                    } catch (Exception exception) {
                        exception.printStackTrace();
                    }
                    break;

            } catch (Exception e) {

                break;
```



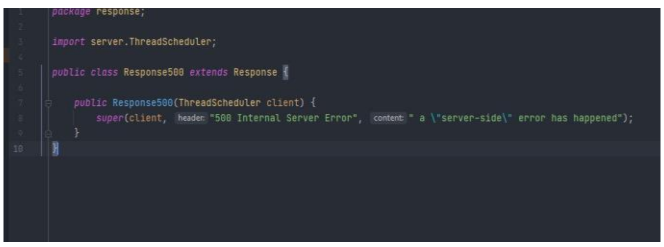
```
import server.ThreadScheduler;

public class Response302 extends Response {

    public Response302(ThreadScheduler client) {
        super(client, header: "302 URL redirection",
                content: " The requested resource has been temporarily moved to a different URI");
    }
}
```



```
import server.ThreadScheduler;

public class Response403 extends Response {

    public Response403(ThreadScheduler client) {
        super(client, header: "403 Forbidden",
                content: " Access to the requested resource is forbidden.");
    }
}
```



```
import server.ThreadScheduler;

public class Response404 extends Response {

    public Response404(ThreadScheduler client) {
        super(client, header: "404 Not Found", content: "The requested file or page is not accessible.");
    }
}
```



```
package response;

import server.ThreadScheduler;

public class Response500 extends Response {

    public Response500(ThreadScheduler client) {
        super(client, header: "500 Internal Server Error", content: " a \"server-side\" error has happened");
    }
}
```

2.1 Discussion

This problem is updating the previous implementation in the code which was only supporting the HTTP response known as 200 Ok. However, in this task, we are required to add support for the 302, 403, 404, and 500 response times.

- ✓ 302

This response is basically about an HTTP response stating that the requested resource has been redirected to a different URL.

This response has been handled by implementing a class called Response302 which will extend the class Response.

Also, a method which is known as displayError302 in the ResponseInvoker class will be called in ThreadScheduler class which will catch exceptions that are known as NoSuchFieldException and RuntimeException which will cause 302 response.

- ✓ 403

This response is basically about the HTTP status indicating that the requested resource is not accessible (forbidden).

This response has been handled by implementing a class called Response403 which will extend the class Response.

Moving on, a method called displayError403Forbidden in the ResponseInvoker class will be called in the ThreadScheduler which will catch an exception known as security exception which will cause a 403 response.

- ✓ 404

This response is basically about the HTTP status indicating that the server was not able to find the requested website.

This response has been handled by implementing a class called Response404 which will extend the class Response.

Consequently, a method called displayError404NotFound in the ResponseInvoker class will be called in the ThreadScheduler which will catch an exception known as FileNotFoundException which will cause a 404 response.

- ✓ 500

This response is basically about the HTTP status indicating that there is a problem on the website's server. However, the problem will not be shown in detail informing that what has caused this problem.

This response has been handled by implementing a class called Response500 which will extend the class Response.

As a result, a method called Display500InternalSeverError in the ResponseInvoker class will be called in the ThreadScheduler which will catch exceptions known as IOException and NullPointerException which will cause 500 response.

Also since it was challenging for us to test this method due to the reason that our sever was working flawlessly, we have intentionally used the following code in the RequestValidator class to throw an IOException and NullPointerException:

"if (header.toString().isEmpty())throw new IOException();"

2.2 VG 2

2.2.1 Discussion

3 Work Distribution

3.1 Disscution

Since this project could have be done in group of two people, we have chosen our team members to be able to finish this project. One of the merits that this group has is that all the team members are familiar with each other. As a result, having this advantage made the working environment pleasant and enjoyable for us.

Regarding the process of completing this project together, a channel has been created in the channel.Consequently, we were able to share our screens to show our works and talk to each other at the same time.

The way how we did the tasks was that two of us have worked on the same question together at the same time. However, sometimes we were stuck at one task and since we did not want to waste our time on one task, we would have asked one of the members to move on and work on the next question. In that case, we would not stay behind, and we would be able to finish all the tasks. Therefore, the two members have worked equally during this project. (50% each).

Additionally, even for writing the report, we have discussed with each other what data, and attachments we must include in this report. All in all, it was very convenient for all of us to work with each other.

Also, at some points, the pictures will indicate that some actions were done by a different member. What I mean is that not all the pictures were taken from the host system of only one member. On the other hand, this report and the pictures are the results of working as a team.